What is claimed is:

1. A method for generating a pipelined synchronized circuit representation of a program loop, comprising:

generating a dependence graph from the program loop, the dependence graph representing operations and registers and connections therebetween;

determining a minimum clock period and initiation interval from the dependence graph, and establishing a current minimum clock period as a function of the minimum clock period and establishing the initiation interval as a current initiation interval; and

iteratively attempting to generate a scheduled graph from operations and registers of the dependence graph using the current minimum clock period and the current initiation interval as scheduling constraints, and if an attempt to generate a scheduled graph fails in an iteration, then increasing the current minimum clock period for a next iteration.

2. The method of claim 1, further comprising:

if attempts to generate a scheduled graph fail with increasing current minimum clock periods, then,

resetting the current minimum clock period to the minimum clock period;

increasing the current initiation interval; and

repeating the step of iteratively attempting to generate a scheduled graph and the steps of resetting the current minimum clock period and increasing the current initiation interval until a scheduled graph is generated.

3. The method of claim 2, further comprising compacting the scheduled graph.

4. The method of claim 3, wherein the scheduled graph is divided into a plurality of time-ordered timeslots that are bounded by storage elements, the method further comprising selectively moving operations between timeslots as a function of storage element requirements of possible movements.

5. The method of claim 4, further comprising inserting registers between selected operations in the scheduled graph.

6. The method of claim 5, further comprising if the initiation interval is greater than 1, generating a controller for selectively enabling the operations and registers in the timeslots.

7. The method of claim 1, further comprising:

converting the program loop to static single assignment (SSA) form;

generating array flow-dependence annotations of the SSA form of the program loop;

converting control flow of the SSA form to data flow graph with predicates; and

synthesizing the data flow graph into the dependence graph.

8. The method of claim 1, wherein increasing the current minimum clock period includes multiplying the current minimum clock period by a selected value.

9. The method of claim 1, wherein the initial value of the current minimum clock period is obtained by multiplying the minimum clock period by a selected dilation factor that is greater than 1.

10. A method for generating a plurality of pipelined synchronized circuit representations of a program loop, comprising:

(a) generating a dependence graph from the program loop, the dependence graph representing operations and registers and connections therebetween;

(b) determining a minimum clock period and initiation interval from the dependence graph and establishing a current minimum clock period by multiplying the minimum clock period by a selected dilation factor and establishing the initiation interval as a current initiation interval;

(c) iteratively attempting to generate a scheduled graph from operations and registers of the dependence graph using the current minimum clock period and the current initiation interval as scheduling constraints, and if an attempt to generate a scheduled graph fails in an iteration, then increasing the current minimum clock period for a next iteration; and

(d) repeating steps (b) - (c) using a plurality of different dilation factors.

11. The method of claim 10, further comprising:

if attempts to generate a scheduled graph fail with increasing current minimum clock periods, then,

resetting the current minimum clock period to the minimum clock period;

increasing the current initiation interval; and

repeating the step of iteratively attempting to generate a scheduled graph and the steps of resetting the current minimum clock period and increasing the current initiation interval until a scheduled graph is generated.

12. The method of claim 11, further comprising compacting the scheduled graph.

13. The method of claim 12, wherein the scheduled graph is divided into a plurality of time-ordered timeslots that are bounded by storage elements, the method further comprising selectively moving operations between timeslots as a function of storage element requirements of possible movements.

14. The method of claim 13, further comprising inserting registers between selected operations in the scheduled graph.

15. The method of claim 14, further comprising if the initiation interval is greater than 1, generating a controller for selectively enabling the operations and registers in the timeslots.

16. The method of claim 10, further comprising:

converting the program loop to static single assignment (SSA) form;

generating array flow-dependence annotations of the SSA form of the program loop;

converting control flow of the SSA form to data flow graph with predicates; and

synthesizing the data flow graph into the dependence graph.

17. The method of claim 10, wherein increasing the current minimum clock period includes multiplying the current minimum clock period by a selected value.

18. An apparatus for generating a pipelined synchronized circuit representation of a program loop, comprising:

means for generating a dependence graph from the program loop, the dependence graph representing operations and registers and connections therebetween;

means for determining a minimum clock period and initiation interval from the dependence graph and establishing a current minimum clock period as a function of the minimum clock period and establishing the initiation interval as a current initiation interval; and

means for iteratively attempting to generate a scheduled graph from operations and registers of the dependence graph using the current minimum clock period and the current initiation interval as scheduling constraints, and if an attempt to generate a scheduled graph fails in an iteration, then increasing the current minimum clock period for a next iteration.

19. An article of manufacture, comprising:

a computer-readable medium configured with program code for causing a computer to perform the steps of,

generating a dependence graph from the program loop, the dependence graph representing operations and registers and connections therebetween;

determining a minimum clock period and initiation interval from the dependence graph, and establishing a current minimum clock period as a function of the minimum clock period and establishing the initiation interval as a current initiation interval; and

iteratively attempting to generate a scheduled graph from operations and registers of the dependence graph using the current minimum clock period and the current initiation interval as scheduling constraints, and if an attempt to generate a scheduled graph fails in an iteration, then increasing the current minimum clock period for a next iteration.